Computational Complexity

21 Jan 2014

Last Time

- Turing machines
- Languages
- Universal Turing Machine
- Polynomial time algorithms and class $\mathbf{P}$

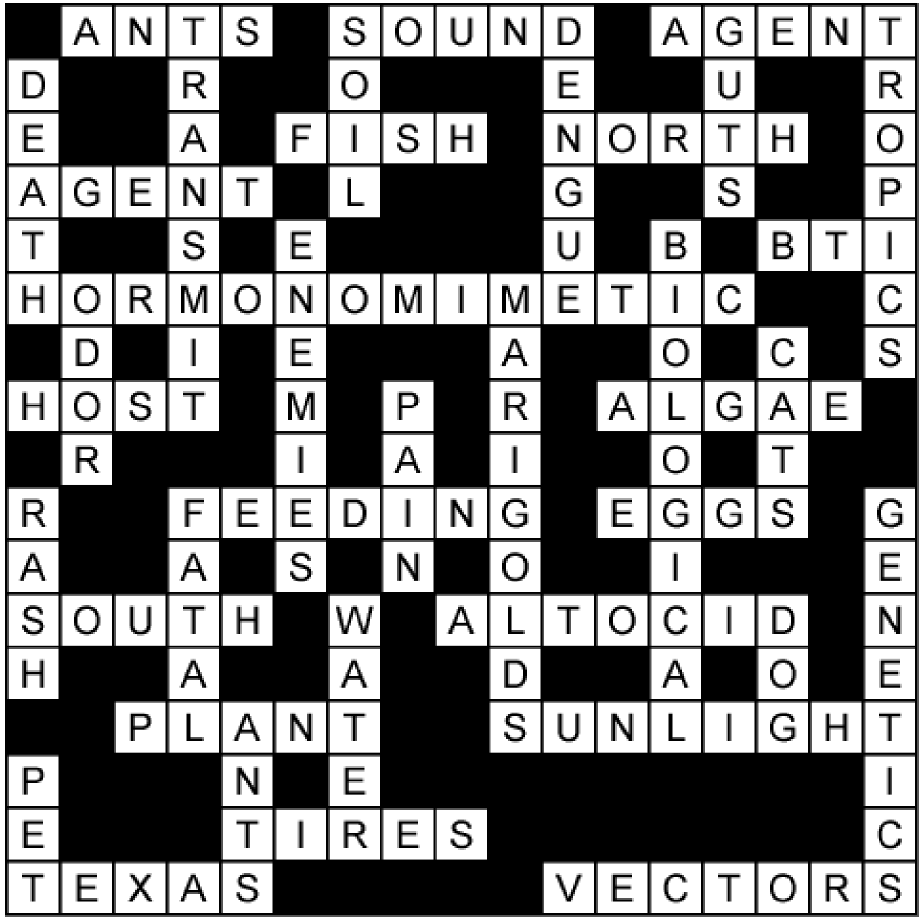
Today

- The complexity class $\mathbf{NP}$ (**N**on-deterministic **P**olynomial time)
- Notion of reduction (way to compare hardness of problems)
- NP completeness (one of the most important notions of complexity)

Crossword Puzzle

- Hard to solve
- Easy to “verify” solutions

- Other examples
  - Easier to verify a proof
  - Easier to appreciate good art



Other Examples

Ellipsoid Method (

- Does G have a Hamiltonian cycle?
- Is there a vector x such that $\boxed{Ax \leq b}$ for matrix A and vector b ?

Real

$\boxed{\text{Integer}}$

$(x_1, x_2, \dots, x_n)$

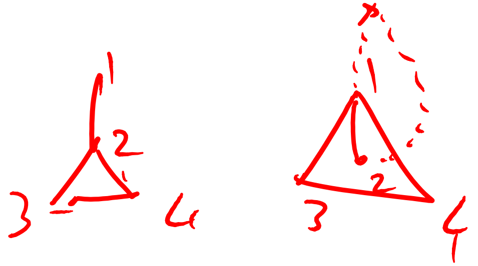
- Is a given number N a composite number? $\boxed{\text{COMP}} \in P$

n

$N = \boxed{N_1 \times N_2}$

- Are graphs G_1 and G_2 isomorphism?

ISO $\overset{?}{\in}$ P



Suppose $\underbrace{G_1 \cong G_2}_{X}$: what is the set of witnesses for $G_1 \cong G_2$

Definition 2.1 *(The class* **NP***)*
A language $L \subseteq \{0,1\}^*$ is in **NP** if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time TM M (called the *verifier* for L) such that for every $x \in \{0,1\}^*$,

$$x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)} \text{ s.t. } \boxed{M(x,u) = 1}\,.$$

L

$$W_x = \left\{ u : M(x,u) = 1 \right\}$$

If $x \in L$ and $u \in \{0,1\}^{p(|x|)}$ satisfy $M(x,u) = 1$ then we call u a *certificate* for x (with respect to the language L and machine M).

- Certificate is also called "witness"

Is $\mathbf{P} = \mathbf{NP}$?

- One of the biggest open questions in math and sciences

- Hundreds of important problems are in **NP** but not known to be in **P** (most of them special cases of integer programming)

- If $\mathbf{P} = \mathbf{NP}$ then all these problems would be easy (solvable in polynomial time)

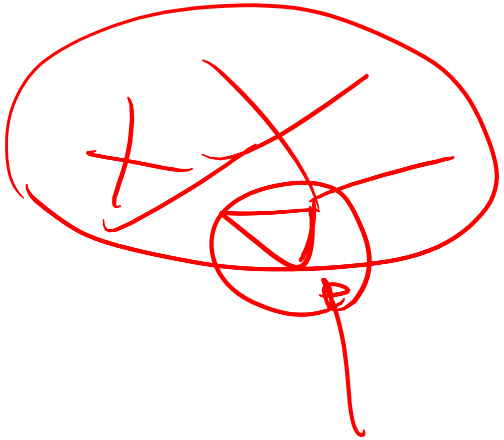
- How can we talk about “relative hardness” of two problems?

Reductions

Which one is harder to solve?

- IND-SET: Given (G, k), does graph G has independent set of size k ?
- CLIQUE: Given (G, k), does graph G has a clique set of size k ?

IND-SET $\geq$ $\leq$ Clique.



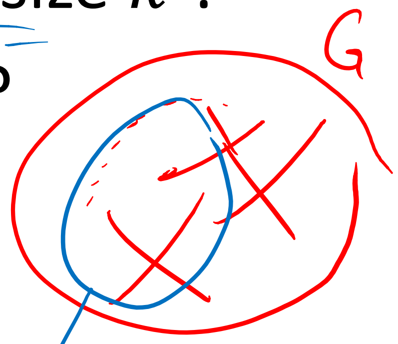
$\boxed{\text{if sblve clique}} \Longrightarrow$ I can sole IND-SET

given (G, k) change G by flipping all edges, H solve (H, k) for Cliqe problem.

Another example

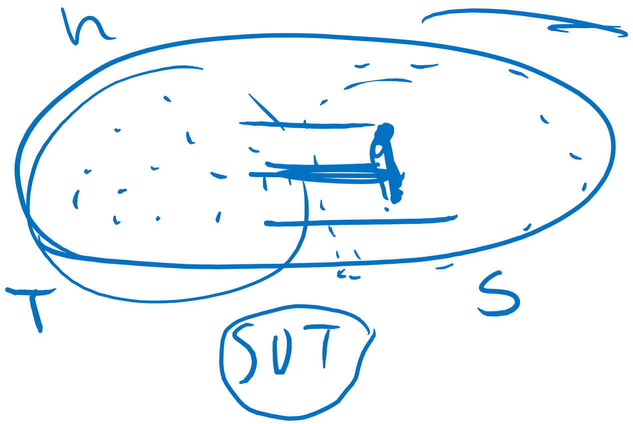
Which one is harder to solve?

- IND-SET: Given (G, k), does graph G has independent set of size k ?
- Set-Cover ~~CLIQUE~~: Given (G, s), does graph G has a set-cover of size s?



IND-SET $\leq$ CLIQUE

given $x = (G, k)$ Q: $(G, k) \in$ IND-SET ?



T is a IND-set iff $\bar{T} = S$ is a set cover

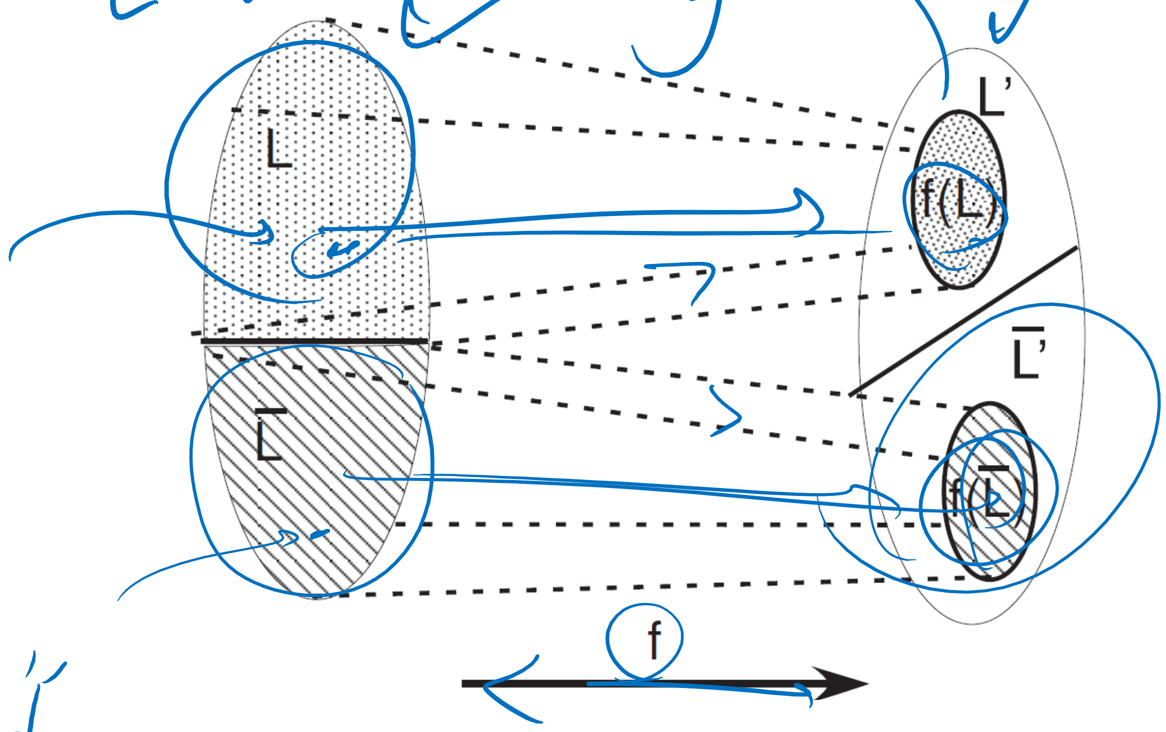
take $x' = (G, n-k)$

$x \in$ IND-set iff $x' \in$ SET-Cover

S is a set cover if all edge (a,b) either $a \in S$ or $b \in S$

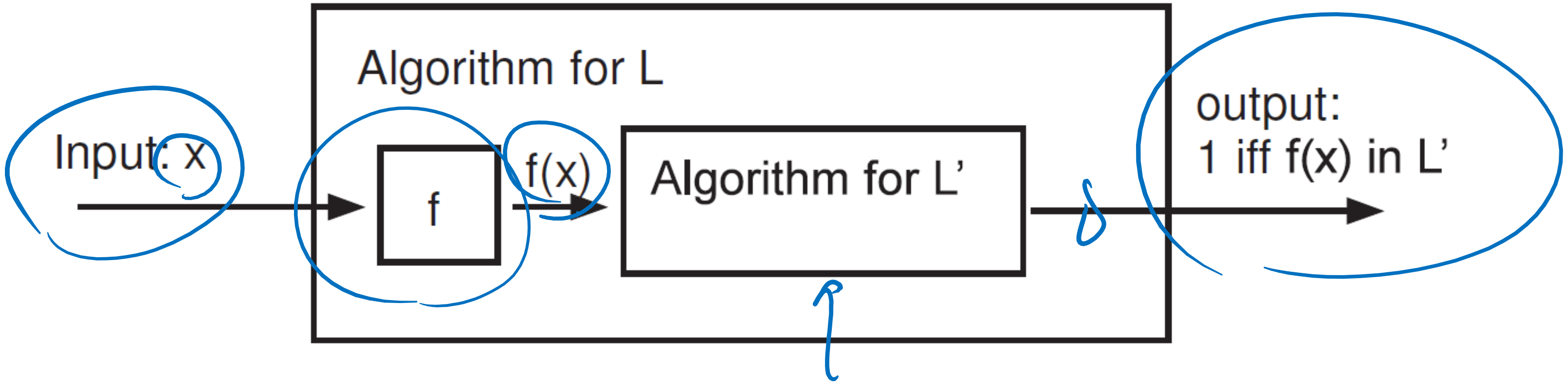
L = IND-set

Clique = L'



L is "reduced" to L'

$L \leq_P L'$



NP hardness and NP completeness

Definition 2.7 *(Reductions,* **NP***-hardness and* **NP***-completeness)*
A language $L \subseteq \{0,1\}^*$ is *polynomial-time Karp reducible to a language* $L' \subseteq \{0,1\}^*$ (sometimes shortened to just "polynomial-time reducible"), denoted by $L \leq_p L'$, if there is a polynomial-time computable function $f: \{0,1\}^* \to \{0,1\}^*$ such that for every $x \in \{0,1\}^*$, $x \in L$ if and only if $f(x) \in L'$.

We say that L' is **NP**-*hard* if $L \leq_p L'$ for every $L \in$ **NP**. We say that L' is **NP**-*complete* if L' is **NP**-hard and $L' \in$ **NP**.

- We can define other "more powerful" reductions as well.

Theorem 2.8

1. (Transitivity) *If* $L \leq_p L'$ *and* $L' \leq_p L''$*, then* $L \leq_p L''$.
2. *If language* L *is* **NP**-*hard and* $L \in \mathbf{P}$ *then* $\mathbf{P} = \mathbf{NP}$.
3. *If language* L *is* **NP**-*complete then* $L \in \mathbf{P}$ *if and only if* $\mathbf{P} = \mathbf{NP}$.

Is there any $\mathbf{NP}$ complete problem?

Theorem 2.9 *The following language is* $\mathbf{NP}$*-complete:*

$$\mathsf{TMSAT} = \{\langle \alpha, x, 1^n, 1^t \rangle : \exists u \in \{0,1\}^n \ \textit{s.t. } M_\alpha \textit{ outputs } 1 \textit{ on input } \langle x, u \rangle \textit{ within } t \textit{ steps}\}$$

where M_α denotes the (deterministic) TM represented by the string α.[2] $\diamondsuit$

$\mathbf{NP}$: Nondeterministic Polynomial Time

NDTM has *two* transition functions δ_0 and δ_1, and a special state denoted by q_{accept}. When an NDTM M computes a function, we envision that at each computational step M makes an arbitrary choice as to which of its two transition functions to apply. For every input x, we say that $M(x) = 1$ if there *exists* some sequence of these choices (which we call the *non-deterministic choices* of M) that would make M reach q_{accept} on input x.

Definition 2.5 For every function $T : \mathbb{N} \to \mathbb{N}$ and $L \subseteq \{0,1\}^*$, we say that $L \in \mathbf{NTIME}(T(n))$ if there is a constant $c > 0$ and a $c \cdot T(n)$-time NDTM M such that for every $x \in \{0,1\}^*$, $x \in L \Leftrightarrow M(x) = 1$ $\diamondsuit$

Theorem 2.6 $\mathbf{NP} = \cup_{c \in \mathbb{N}} \mathbf{NTIME}(n^c)$

Next Time

- Boolean Satisfiability Problem is NP-Hard (Cook-Levin Theorem)
- Many Interesting Combinatorial Problems are NP-Hard (Karp)
- Search Problems vs. Decision Problems
- Philosophical Impacts of $\mathbf{P} = \mathbf{NP}$ and $\mathbf{P} \neq \mathbf{NP}$